In the chat, please share:

- Where and what you teach
- How you’re teaching
  - Face-to-face
  - Hybrid
  - Fully remote
DOLE WHIP CS:
Creating Engaging Learning Experiences for All

A presentation prepared by Omar Shepherd
Community Ambassador, CS for All Teachers
January 27, 2021
Housekeeping

Participants can use the controls at the bottom of the Zoom window to:
● Mute/unmute and switch between computer and phone audio
● Share/stop webcams
● Access the participant list
● Chat with fellow attendees, panelists, and hosts
● Leave the meeting
● . . . and more!
Housekeeping

- Use reactions found in the attendee controls
- Participants can display thumbs up or clapping hands over webcam or name
- Reactions will disappear after 5 seconds

- Additional nonverbal feedback options are available under the Participants button
- Icons under the more button are agree, disagree, clap, need a break, and away
- Only one symbol displays at a time
- Click the icon again to remove it
Housekeeping

- Switch between Gallery and Speaker Views when webcams are shared in upper right corner

- Adjust view of the meeting and access annotate feature under View Options menu at top of Zoom window
Virtual Meeting/Conference
Recording Notice

The American Institutes for Research (AIR) allows for the recording of audio, visuals, participants, and other information sent, verbalized, or utilized during business related meetings. By joining a meeting, you automatically consent to such recordings. Any participant who prefers to participate via audio only should disable their video camera so only their audio will be captured. Video and/or audio recordings of any AIR session shall not be transmitted to an external third party without their permission.
Introductions

Omar Shepherd

Curriculum Specialist
STEM/Career Education (CTE)
Email: oshepherd@ocde.us
Twitter: @doctorstem
Goals

1. Explore the DOLE WHIP lesson planning strategy to engage diverse learners in computer science.

2. Review a sample lesson and consider resources for implementation.
High Tech High
Project-Based Demonstration School
High Tech High - PBL Demonstration School

- Lesson plan strategy developed during High Tech High PBL Leadership Academy

- Click here to view student projects
High Tech High - San Diego, CA
High Tech High - San Diego, CA
High Tech High - San Diego, CA
WELCOME TO
TEAM MccAmMACK/WAKEFIELD'S
COMIC BOOK EXHIBITION

PHYSICS ESSENTIAL QUESTIONS:
- HOW CAN YOU USE PHYSICS TO CREATE SUPER POWERS?
- HOW CAN YOU BALANCE REAL WORLD PHYSICS WITH COMIC BOOK
  FICTIONAL ELEMENTS?

HUMANITIES ESSENTIAL QUESTIONS:
- HOW CAN YOU DEVELOP A STRUCTURED PLOT LINE WITH DEEP CHARACTERS
  THAT THE AUDIENCE WILL CARE ABOUT?
- WHAT HAS SOCIETY ACCEPTED AS HEROIC AND VILLAINOUS QUALITIES?
- HOW CAN YOU CREATE A CHARACTER THAT CHALLENGES STATUS QUO
  BELIEFS ABOUT HEROIC AND VILLAINOUS QUALITIES?

STUDENTS SHOULD BE PROUD TO SHOW YOU THEIR COMIC, THEIR PROJECT
FOLDER, AND A HANDS ON PHYSICS DEMONSTRATION. ENJOY!
High Tech High - San Diego, CA
CONIC SECTIONS
Equation for an ellipse: \( \frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \)

Where cutting a cone, depending on the angle, the edges will form a
cone, ellipse, parabola, or hyperbola. This graph is set 8 in. in resemblance
to an ellipse-type shape. With this equation, you can find the
specific points for x and y that the equality above holds.
Staircase to Nowhere
High Tech High - San Diego, CA
High Tech High - San Diego, CA
High Tech High - San Diego, CA
Lesson Planning Strategy
DOLE WHIP CS Lesson Planning

D - Delivering
O - Outstanding
L - Learning
E - Experiences

W - Wildy Engaging
H - High Impact
I - Instant Feedback
P - Pedagogy
What do you want your students to create, build or do?

What's the Project Seed?

Who is the authentic audience?

What will students learn?

Where will your products "live"?
What's the Project Purpose?

Why do these lessons matter?

What will make students excited to come to school?

Why will your students care?

How will this project make a positive difference?
What will be your entry point?
What is your driving question?

Is it teacher driven?  
Is it student driven?
How do you see your students engaging in these STEM practices?
Where do you see opportunities to incorporate "Future Ready" skills?

Critical Thinking

Creativity

Communication

Collaboration

Character

Content Goals

What specific academic content will the students learn?

How?
What will students READ?
Where are there opportunities for differentiation?

Special Needs... ELL...
Challenge me...
Shy... Social...
Use these reflective questions at any time to reflect on the design implementation of these educational experiences.

**Voice and Choice**
How are these experiences designed to include student, parent, and community voice in the co-design of educational experiences?

**Equity and Diversity**
How are these experiences designed to provide access and challenge for all students to engage in meaningful work?

How are these experiences designed to value a wide range of perspectives, skills, knowledge, content, and products?

**Reflective Practice**
How are these experiences designed to foster thoughtful and deliberate practice?

**Passion**
How are these experiences designed to tap into students' and teachers' personal questions, values, and passions?
<table>
<thead>
<tr>
<th>Access</th>
<th>Provide options for Recruiting Interest</th>
<th>Provide options for Perception</th>
<th>Provide options for Physical Action</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Give students choices (choose project, software, topic)</td>
<td>• Model computing using physical representations as well as through an interactive whiteboard, videos</td>
<td>• Provide teacher’s codes as templates</td>
</tr>
<tr>
<td></td>
<td>• Allow students to make projects relevant to culture and age</td>
<td>• Give access to modeled code while students work independently</td>
<td>• Include CS Unplugged activities that show physical relationship of abstract computing concepts</td>
</tr>
<tr>
<td></td>
<td>• Minimize possible common “pitfalls” for both computing and content</td>
<td>• Provide access to video tutorials of computing tasks</td>
<td>• Use assistive technology including larger/smaller mice, touch-screen devices</td>
</tr>
<tr>
<td></td>
<td>• Allow for differences in pacing and length of work sessions</td>
<td>• Select coding apps and websites that allow the students to adjust visual settings (such as font size &amp; contrast) and that are compatible with screen readers.</td>
<td>• Select coding apps and websites that allow coding with keyboard shortcuts in addition to dragging &amp; dropping with a mouse</td>
</tr>
<tr>
<td>Multiple Means of Engagement</td>
<td>Multiple Means of Representation</td>
<td>Multiple Means of Action &amp; Expression</td>
<td></td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------------</td>
<td>--------------------------------------</td>
<td></td>
</tr>
<tr>
<td><strong>Build</strong></td>
<td><strong>Provide options for</strong></td>
<td><strong>Provide options for</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Sustaining Effort &amp; Persistence</strong></td>
<td><strong>Language &amp; Symbols</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Remind students of both computing and content goals</td>
<td>Teach and review content specific vocabulary</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Provide support or extensions for students to keep engaged</td>
<td>Teach and review computing vocabulary (e.g., code, animations, computing, algorithm)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Teach and encourage peer collaboration by sharing products</td>
<td>Post anchor charts and provide reference sheets with images of blocks or with common syntax when using text</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Utilize pair programming and group work with clearly defined roles</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Discuss the integral role of perseverance and problem solving in computer science</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Recognize students for demonstrating perseverance and problem solving in the classroom</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Give options of unplugged activities and computing software and materials (e.g., Pseudocode, Scratch, code.org, Alice)</td>
<td>Give opportunities to practice computing skills and content through projects that build prior lessons</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Provide sentence starters or checklists for communicating in order to collaborate, give feedback, and explain work</td>
<td>Provide options that include starter code</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Create physical manipulatives of commands, blocks or lines of code</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Multiple Means of Engagement</th>
<th>Multiple Means of Representation</th>
<th>Multiple Means of Action &amp; Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Affective Networks</strong></td>
<td><strong>Recognition Networks</strong></td>
<td><strong>Strategic Networks</strong></td>
</tr>
<tr>
<td>The “WHY” of learning</td>
<td>The “WHAT” of learning</td>
<td>The “HOW” of learning</td>
</tr>
<tr>
<td><strong>Internalize</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Provide options for</td>
<td>Provide options for</td>
<td>Provide options for</td>
</tr>
<tr>
<td><strong>Self Regulation</strong></td>
<td><strong>Comprehension</strong></td>
<td><strong>Executive Functions</strong></td>
</tr>
<tr>
<td>• Communicate clear</td>
<td>• Activate background knowledge</td>
<td>• Guide students to set goals for</td>
</tr>
<tr>
<td>expectations for computing</td>
<td>by making computing tasks</td>
<td>long-term projects</td>
</tr>
<tr>
<td>tasks, collaboration, and</td>
<td>interesting and culturally</td>
<td>• Record students’ progress (have</td>
</tr>
<tr>
<td>help seeking</td>
<td>relevant</td>
<td>planned checkpoints during lessons</td>
</tr>
<tr>
<td>• Develop ways for students</td>
<td>• State lesson content/ computing</td>
<td>for understanding and progress for</td>
</tr>
<tr>
<td>to self-assess and reflect</td>
<td>goals</td>
<td>computing skills and content)</td>
</tr>
<tr>
<td>on own projects and those</td>
<td>• Encourage students to ask</td>
<td>• Provide exemplars of completed</td>
</tr>
<tr>
<td>of others</td>
<td>questions as comprehension</td>
<td>products</td>
</tr>
<tr>
<td>• Use assessment rubrics</td>
<td>checkpoints</td>
<td>• Embed prompts to stop and plan,</td>
</tr>
<tr>
<td>that evaluate both content</td>
<td>• Use relevant analogies and</td>
<td>test, or debug throughout a lesson</td>
</tr>
<tr>
<td>and process</td>
<td>make cross-curricular connections</td>
<td>or project.</td>
</tr>
<tr>
<td>• Break-up coding activities</td>
<td>explicit (for example comparing</td>
<td>• Provide graphic organizers to</td>
</tr>
<tr>
<td>with opportunities for</td>
<td>iterative product development to</td>
<td>facilitate planning, goal-setting,</td>
</tr>
<tr>
<td>reflection such as turn and</td>
<td>the writing process)</td>
<td>and debugging</td>
</tr>
<tr>
<td>talks or written questions</td>
<td>• Provide graphic organizers for</td>
<td>• Provide explicit instruction on</td>
</tr>
<tr>
<td>• Acknowledge difficulty</td>
<td>students to “translate” programs</td>
<td>skills such as asking for help,</td>
</tr>
<tr>
<td>and frustration. Model</td>
<td>into pseudocode</td>
<td>providing feedback, and using</td>
</tr>
<tr>
<td>different strategies for</td>
<td></td>
<td>problem solving techniques</td>
</tr>
<tr>
<td>dealing with frustration</td>
<td></td>
<td>• Demonstrate debugging with</td>
</tr>
<tr>
<td>appropriately</td>
<td></td>
<td>think-alouds</td>
</tr>
</tbody>
</table>
Sample Lesson

Topic: Variables and Variable Assignment
# Backwards Planning for DOLE WHIP CS

## Lesson: Assignment and Reassignment of an Object

### Planning Backwards (Part 1): Vision

<table>
<thead>
<tr>
<th>Your Vision or Project “Seed”</th>
<th>What will the students learn or discover?</th>
</tr>
</thead>
<tbody>
<tr>
<td>What do you want students to create, build, or do? List ideas of final products.</td>
<td>- Students will learn that the <code>=</code> sign when writing a program is merely a reference for an object and not an absolute.</td>
</tr>
<tr>
<td>A Python Program:</td>
<td>- Students will discover connections to working with variables in math class.</td>
</tr>
<tr>
<td>Program that demonstrates understanding of:</td>
<td>- Students will learn that in programming variable assignment is not absolute rather, variables can be reassigned and even updated within a program.</td>
</tr>
<tr>
<td>- Variable Assignment in programming</td>
<td></td>
</tr>
<tr>
<td>- Reassignment of Variables</td>
<td></td>
</tr>
<tr>
<td>- Updating Variables within a program</td>
<td></td>
</tr>
</tbody>
</table>

### Audience

<table>
<thead>
<tr>
<th>Who is the authentic audience for your students’ work?</th>
<th>Where will your products “live” after the lessons end?</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Fellow Students in the Classroom and Peers</td>
<td>- Students will submit their lessons into the LMS via link or uploaded PDF or Word document.</td>
</tr>
<tr>
<td>upon share out of experiences.</td>
<td>- Students will create a narrated video of the developed program demonstrating their understanding and some their execution of the program within the context of the</td>
</tr>
</tbody>
</table>

(Click image to access)
Chapter Two: Variables

Assignment, Reassignment, and Updating Variables
What is a Variable?
A variable is a name that refers to a value

name = Value
One of the most powerful features of a programming language is the ability to manipulate variables. A variable is a name that refers to a value.
Variable Assignment

```
message = "What's up, Doc?"
n = 17
pi = 3.14159

print(message)
print(n)
print(pi)
```
Variable Reassignment

The first time `bruce` is printed, its value is 5, and the second time, its value is 7. The assignment statement changes the value (the object) that `bruce` refers to.
In math, a statement of equality is always true. If \( a \) is equal to \( b \) now, then \( a \) will always equal to \( b \).

In Python, an assignment statement can make two variables refer to the same object and therefore have the same value. They appear to be equal. However, because of the possibility of reassignment, they don’t have to stay that way.

```
1 a = 5
2 b = a  # after executing this line, a and b are now equal
3 print(a, b)
4 a = 3  # after executing this line, a and b are no longer equal
5 print(a, b)
```
Question #1:

data-4-1: What is printed when the following statements execute?

```python
day = "Thursday"
day = 32.5
day = 19
print(day)
```
Two-Part Assignment

Part One:
- Draft a program in Python where you assign a value to a variable and subsequently re-assign the variable value within the same program.

Part Two:
- Draft a video or slide deck presentation with screenshots and bullet points describing what’s happening computationally and what changes as the program is manipulated.
Click [here](#) to access DOLE WHIP CS template
Questions to Keep in Mind

1. How would you assess that your students had met the CS standard you were working on?

2. How would you assess if your students had met the content standard you were working on?

3. What would you expect to witness if this practice were active in a classroom?
## Session Links and Resources

(Click image to access)

### DOLE WHIP CS

<table>
<thead>
<tr>
<th>Session Links and Resources</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Session Presentation Link</strong></td>
</tr>
<tr>
<td><a href="https://docs.google.com/presentation/d/1hPgrdmCUA1TTGxzkeeeD4XqGh4JcqnPx4b5Q7WP7zaM/edit?usp=sharing">https://docs.google.com/presentation/d/1hPgrdmCUA1TTGxzkeeeD4XqGh4JcqnPx4b5Q7WP7zaM/edit?usp=sharing</a></td>
</tr>
<tr>
<td><strong>DOLE WHIP Lesson Plan Template</strong></td>
</tr>
<tr>
<td><a href="https://docs.google.com/document/d/1uPiUZ-8i6YrzAqUueOx0lNT5e8uywL4bGcNw7fw59c/copy">https://docs.google.com/document/d/1uPiUZ-8i6YrzAqUueOx0lNT5e8uywL4bGcNw7fw59c/copy</a></td>
</tr>
</tbody>
</table>

#### Computer Science Standards Resources

<table>
<thead>
<tr>
<th>California CS Standards</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="https://docs.google.com/presentation/d/1hPgrdmCUA1TTGxzkeeeD4XqGh4JcqnPx4b5Q7WP7zaM/edit?usp=sharing">Introduction to the K12 CS Standards</a></td>
</tr>
<tr>
<td><a href="https://docs.google.com/document/d/1uPiUZ-8i6YrzAqUueOx0lNT5e8uywL4bGcNw7fw59c/copy">K12 CS Standards</a></td>
</tr>
<tr>
<td><a href="https://docs.google.com/presentation/d/1hPgrdmCUA1TTGxzkeeeD4XqGh4JcqnPx4b5Q7WP7zaM/edit?usp=sharing">K12 CS Standards progressions chart</a></td>
</tr>
<tr>
<td><a href="https://docs.google.com/presentation/d/1hPgrdmCUA1TTGxzkeeeD4XqGh4JcqnPx4b5Q7WP7zaM/edit?usp=sharing">K12 CS Standards appendix</a></td>
</tr>
</tbody>
</table>

#### Resources for Content Integration

<table>
<thead>
<tr>
<th>Content Standards</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="https://docs.google.com/presentation/d/1hPgrdmCUA1TTGxzkeeeD4XqGh4JcqnPx4b5Q7WP7zaM/edit?usp=sharing">CCSS ELA</a></td>
</tr>
<tr>
<td><a href="https://docs.google.com/presentation/d/1hPgrdmCUA1TTGxzkeeeD4XqGh4JcqnPx4b5Q7WP7zaM/edit?usp=sharing">History Social Science</a></td>
</tr>
<tr>
<td><a href="https://docs.google.com/presentation/d/1hPgrdmCUA1TTGxzkeeeD4XqGh4JcqnPx4b5Q7WP7zaM/edit?usp=sharing">VAPA</a></td>
</tr>
</tbody>
</table>
Questions
Upcoming Activities

● Webinar February 23: Micro:bits in the Classroom - Strategies to Use Virtually or In Person (registration coming soon!)
Are you a member of CS for All Teachers?

Membership is FREE!
Sign up in order to:

- Connect with other CS teachers
- Participate in free webinars
- Access free resources

https://csforallteachers.org/user/register